

BAB IV METODE PENELITIAN

4.1 Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Laboratorium Instrumentasi Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya Malang pada tahun 2016.

4.2 Alat dan Bahan

Alat yang digunakan pada penelitian ini antara lain adalah Textronix TDS 1001B *two channel digital storage oscilloscope*, Laptop-PC, *Software ISE Design Suite*, *Software STM32 Emwin GUI Builder*, *Software SW4STM32*, *Software STM32CubeMx*, rangkaian osilator untuk QCM, multimeter digital, dan JTAG HS2 *Programming Cable*. Sedangkan bahan yang digunakan antara lain adalah Modul STM32F7 *Discovery*, *micro SD* 8GB, IC CPLD XC2C512, Sensor QCM 10 MHz, Kristal OCXO AOCJY 100MHz, *Powerbank* 6000 mAh, dan beberapa komponen elektronik.

4.3 Desain Sistem

Desain sistem pengukuran QCM portabel ini secara garis besar ditunjukkan seperti pada Gambar 4.1. Sistem pencacah frekuensi portabel ini dibagi dalam empat sistem utama, yaitu pencacah frekuensi berbasis CPLD, modul mikrokontroller 32 bit berbasis STM32F746NG (STM32F7 *Discovery*), dan system daya atau *power supply*. Sistem-sistem ini akan terintegrasi untuk menghasilkan sebuah perangkat pengukuran QCM portabel yang akan mengukur

frekuensi sensor QCM pada masing-masing *channel* dan menyimpannya pada *micro SD*, serta menampilkannya secara *real-time* pada layar LCD TFT 4.3 inch. Mikrokontroler STM32F746NG merupakan sistem kontrol utama yang akan menangani beberapa fungsi seperti menerima dan mengolah data cacahan yang terukur oleh pencacah frekuensi berbasis CPLD, mengontrol LCD TFT 4.3inch, membaca sensor *Touchscreen* yang terintegrasi pada modul LCD TFT, menangani fungsi *Graphics User Interface*, dan menyimpan data pada *micro SD*.

Perangkat portabel ini didesain agar mampu mengukur frekuensi dari dua sensor QCM secara *real-time* dan simultan. Oleh karena itu, pada penelitian ini akan dikembangkan dua buah pencacah frekuensi yang bekerja secara simultan untuk mengukur frekuensi sensor QCM yang dibangkitkan oleh osilator. Pencacah frekuensi yang dikembangkan pada penelitian ini dibangun menggunakan CPLD XC2C512. Metode yang digunakan untuk membangun pencacah frekuensi ini adalah metode *hybrid* seperti yang dikemukakan pada subbab 2.2. Oleh karena itu, terdapat beberapa desain rangkaian digital yang dibangun di dalam CPLD ini, yaitu Pembagi Frekuensi, Multiplekser, empat Pencacah Digital dan empat *Register* pada masing-masing *Channel*.

Pembagi frekuensi akan mengubah frekuensi osilator eksternal OCXO 100MHz menjadi sinyal *time-gate* satu detik. Sinyal *time-gate* satu detik ini digunakan sebagai gerbang dan durasi proses pencacahan. Selain itu, sinyal *time-gate* satu detik ini juga digunakan sebagai sinyal *interrupt* yang mengontrol proses pengiriman data dari CPLD ke STM32F746NG, pengiriman data dari STM32F746NG ke *micro SD*, dan update LCD TFT. Pembagi frekuensi ini dibangun menggunakan pencacah digital yang akan mengubah frekuensi dari kristal

OCXO100MHz (periode 10 nanodetik). Pencacah digital ini bekerja dengan cara mencacah sinyal *clock* eksternal sebanyak 100000010 cacahan untuk menghasilkan sinyal *high* selama satu detik (100000000 *clock* eksternal) dan sinyal *low* selama 100 nanodetik (10 *clock* eksternal). Sinyal *high* selama satu detik inilah yang digunakan sebagai sinyal *time-gate* atau gerbang dimana proses pencacahan sinyal input dilakukan. Sedangkan 100 nanodetik sinyal *low* dalam penelitian ini disebut sebagai sinyal *time-delay* yang digunakan sebagai sinyal dimana proses penyimpanan data ke *Register* dilakukan.

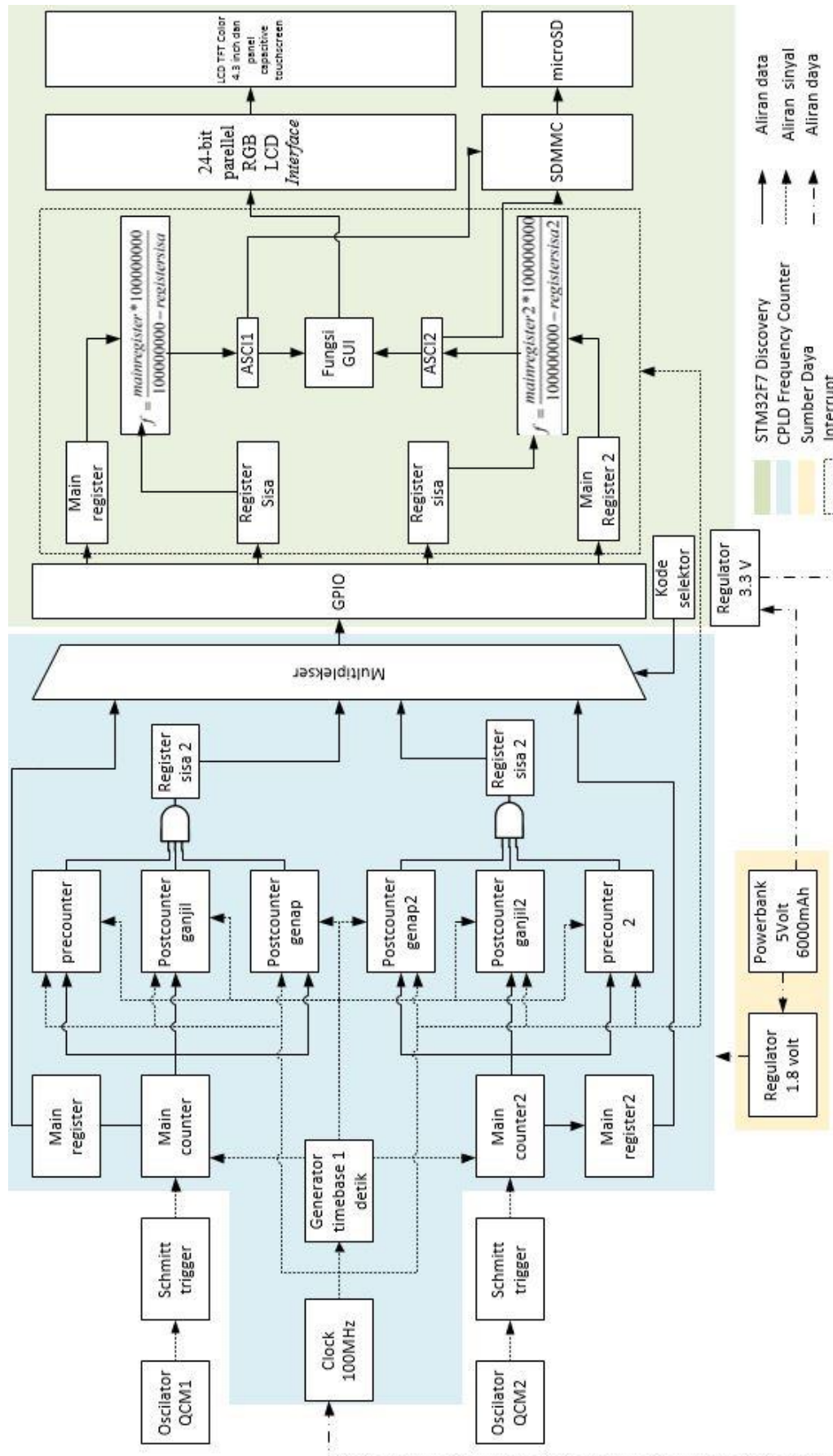
Proses pengukuran dimulai dengan membangkitkan frekuensi resonansi kristal kuarsa menggunakan sebuah rangkaian osilator. Kemudian sinyal ini akan melewati *schmitt trigger* yang berfungsi sebagai filter yang akan membatasi sinyal dengan tegangan rendah yang dianggap sebagai *noise* agar tidak tercacah (Wijayanto, 2014). Setelah melewati *schmitt trigger* maka sinyal akan berubah menjadi sinyal kotak. Kemudian sinyal ini akan dicacah menggunakan pencacah digital (*main counter*) yang dibangun di dalam CPLD. Pencacah digital ini akan mencacah sinyal input selama sinyal *time-gate* satu detik seperti halnya pada pengukuran frekuensi menggunakan metode konvensional. Hasil cacahannya akan disimpan pada *mainRegister* ketika terjadi transisi sinyal *time-gate* satu detik dari level high ke level low. Selain itu, durasi antara transisi *rising-edge* sinyal *time-gate* satu detik dengan transisi *rising-edge* pertama dari sinyal input diukur dengan cara mencacah sinyal *clock* eksternal menggunakan *pracounter* seperti halnya pada pengukuran frekuensi metode *reciprocal*. Sedangkan *pascacountergenjil* dan *pascacountergenap* digunakan untuk mencacah sinyal *clock* eksternal selama durasi terjadinya transisi *rising edge* terakhir dari sinyal input dengan transisi

falling edge dari sinyal *time-gate* satu detik. Hasil cacahannya masing-masing akan disimpan pada *praRegister*, *pascaRegisterganjil*, dan *pascaRegistergenap* pada saat sinyal *time-delay*. Proses cacahan tersebut juga terjadi secara simultan pada proses pengukuran di *channel 2*. Kemudian data pengukuran yang tersimpan pada masing-masing *Register* akan dikirim secara bertahap melalui multiplekser ke mikrokontroller STM32F746NG. Proses pengiriman data terjadi jika mikrokontroller menerima sinyal *interrupt* berupa sinyal *time-gate* satu detik dari CPLD. Sehingga selama sinyal *time-gate* satu detik ini, selain terjadi proses cacahan, juga terjadi proses pengiriman data ke mikrokontroller. Proses pengiriman akan dikontrol oleh kode yang dikirim oleh mikrokontroller STM32F746NG. Data akan dikirim secara parallel dengan setiap paket data berukuran 16 bit.

Pada mikrokontroller STM32F746NG ini data pengukuran disimpan di masing-masing *Register* yang bersesuaian dengan hasil cacahan yang dilakukan oleh CPLD. Setelah itu, dilakukan proses perhitungan untuk mendapatkan frekuensi resonansi terukur dari sensor QCM seperti yang tampak pada Gambar 4.1. Kemudian, hasil perhitungan tersebut dikonversi ke dalam bentuk kode ASCII agar lebih mudah dalam proses pengiriman ke PC dan penyimpanan di micro SD. Selanjutnya, data ini akan dikirim dan disimpan di micro SD serta ditampilkan pada LCD TFT setelah melewati fungsi GUI.

Perangkat pencacah frekuensi yang dikembangkan pada penelitian ini bersifat portabel dan mobile. Sehingga dibutuhkan sumber daya yang terintegrasi langsung dengan perangkat tersebut. Selain itu, sumber daya tersebut memiliki kapasitas yang memungkinkan perangkat pencacah frekuensi portabel tersebut dapat digunakan hingga 12 jam *non-stop*. Oleh karena itu, sumber daya utama yang

digunakan pada perangkat ini berasal dari *Powerbank 6000 mAh* dengan output 5V. Selain itu, *power supply* ini juga dirancang agar mampu menghasilkan empat sumber tegangan stabil 3.3volt, 3.3volt/1A, dan 1.8volt. Tegangan 5volt digunakan untuk mencatu IC *schmitt-trigger*, 3.3volt untuk mencatu STM32F407VG dan pin I/O pada IC CPLD, 3.3volt/1A untuk mencatu modul LCD TFT *Touchscreen*, dan 1.8volt untuk mencatu IC CPLD.



Gambar 4.1 Desain sistem pencacah frekuensi portabel untuk sensor QCM

4.3.1 Desain Program

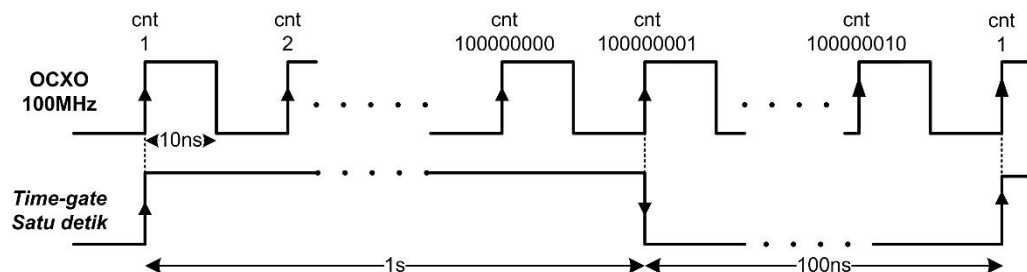
Secara garis besar penelitian ini menggunakan dua komponen utama yang akan menangani fungsi sesuai dengan program yang ditanamkan ke dalamnya. Kedua komponen tersebut adalah CPLD XC5C512 dan modul mikrokontroller STM32F7 *Discovery* berbasis mikrokontroller STM32F746NG. Program yang ditanamkan pada CPLD akan memberikan kemampuan pada CPLD agar dapat berfungsi sebagai pencacah frekuensi dua kanal yang masing-masing dapat mencacah hingga 26 bit dengan resolusi 0.1Hz. Sedangkan program yang ditanamkan pada mikrokontroller STM32F746NG akan memberikan kemampuan pada mikrokontroller ini agar dapat mengontrol dan berkomunikasi dengan CPLD dan LCD TFT *Touchscreen*.

4.3.1.1 CPLD Frequency Counter

Program untuk CPLD dibangun menggunakan *software Xilinx ISE Design Suite* dengan bahasa pemrograman VHDL (*Very High Speed Integrated Circuit Hardware Description Language*). Secara garis besar, terdapat empat desain program yang dibangun di dalam CPLD yaitu program pembagi frekuensi, *digital counter*, *Register*, dan multiplekser (Wijayanto, 2014). Pencacah frekuensi yang dikembangkan pada penelitian ini merupakan pengembangan dari pencacah frekuensi metode *hybrid*. Alur program pencacah frekuensi yang dibangun pada CPLD ini dapat ditunjukkan seperti pada Gambar 4.4. Program pencacah frekuensi yang dikembangkan ini terdiri dari beberapa fungsi untuk membangun sebuah pencacah frekuensi dua *channel* dengan metode hibrid. Fungsi-fungsi tersebut adalah fungsi *time-gate generator*, *main-counter*, *pre-counter*, *post-counter*,

restore data ke Register, dan fungsi *transfer data ke STM32F746NG*. Setiap fungsi yang dibangun dalam program ini bekerja secara paralel, simultan, dan terintegrasi. Program dimulai dengan menginisialisasi *Port*, *Library*, dan *Architecture* yang digunakan dalam program.

Time-gate generator digunakan untuk membangkitkan sinyal *time-gate* satu detik yang digunakan sebagai gerbang dimana proses pencacahan dilakukan. Fungsi ini dijalankan dengan cara membangun sebuah pencacah digital yang akan mencacah sinyal osilator OCXO 100MHz dimana output dari pencacah digital ini merupakan sinyal gelombang penuh yang memiliki periode 1.0000001 detik. Sinyal ini memiliki kondisi *high* atau bernilai 1 selama 1 detik dan kondisi *low* atau bernilai 0 selama 100 nanodetik seperti pada Gambar 4.2. Kondisi *high* selama satu detik inilah yang disebut sebagai sinyal *time-gate*, sedangkan kondisi *low* selama 100 nano detik ini disebut sebagai sinyal *time-delay*.

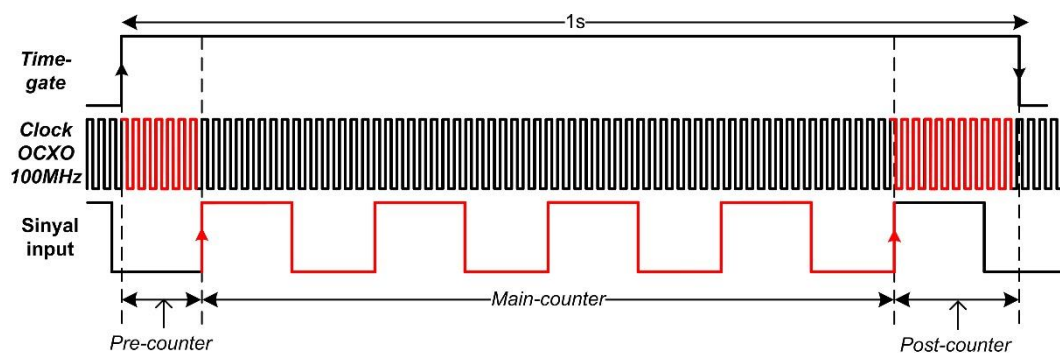


Gambar 4.2 Pembentukan sinyal *time-gate* satu detik

Periode sinyal *clock* eksternal yang digunakan sebesar 10 ns, sehingga untuk membentuk sinyal output pencacah digital dengan periode 1.0000001 detik maka pencacah digital tersebut harus mencacah sinyal *clock* eksternal sebanyak 100000010 kali. Output pencacah digital akan dipertahankan dalam kondisi high atau bernilai 1 (*gate*=1) selama 100ribu cacahan pertama untuk membentuk sinyal *time-gate* selama satu detik. Pada cacahan ke 100ribu, sinyal output dari pencacah

digital akan mengalami *falling-edge* dan selanjutnya akan dipertahankan dalam kondisi *low* atau bernilai 0 ($\text{gate}=0$) selama 10 cacahan berikutnya untuk membentuk sinyal *time-delay* selama 100ns. Selanjutnya sinyal ini akan mengalami *rising-edge* dan *digital counter* akan direset pada saat cacahan ke 100000010.

Terdapat dua *channel* pengukuran QCM yang masing-masing *channel* dapat mengukur frekuensi QCM hingga 67 MHz atau setara dengan 26 bit. Setiap *channel* mengukur frekuensi dengan metode *hybrid*. Oleh karena itu, pada program yang ditanamkan pada CPLD terdapat tiga fungsi yang digunakan untuk membangun pencacah frekuensi dengan metode *hybrid*, yaitu fungsi *main-counter*, *pre-counter*, dan *post-counter* seperti pada Gambar 4.3.



Gambar 4.3 Metode hibrid untuk pencacah frekuensi sensor QCM

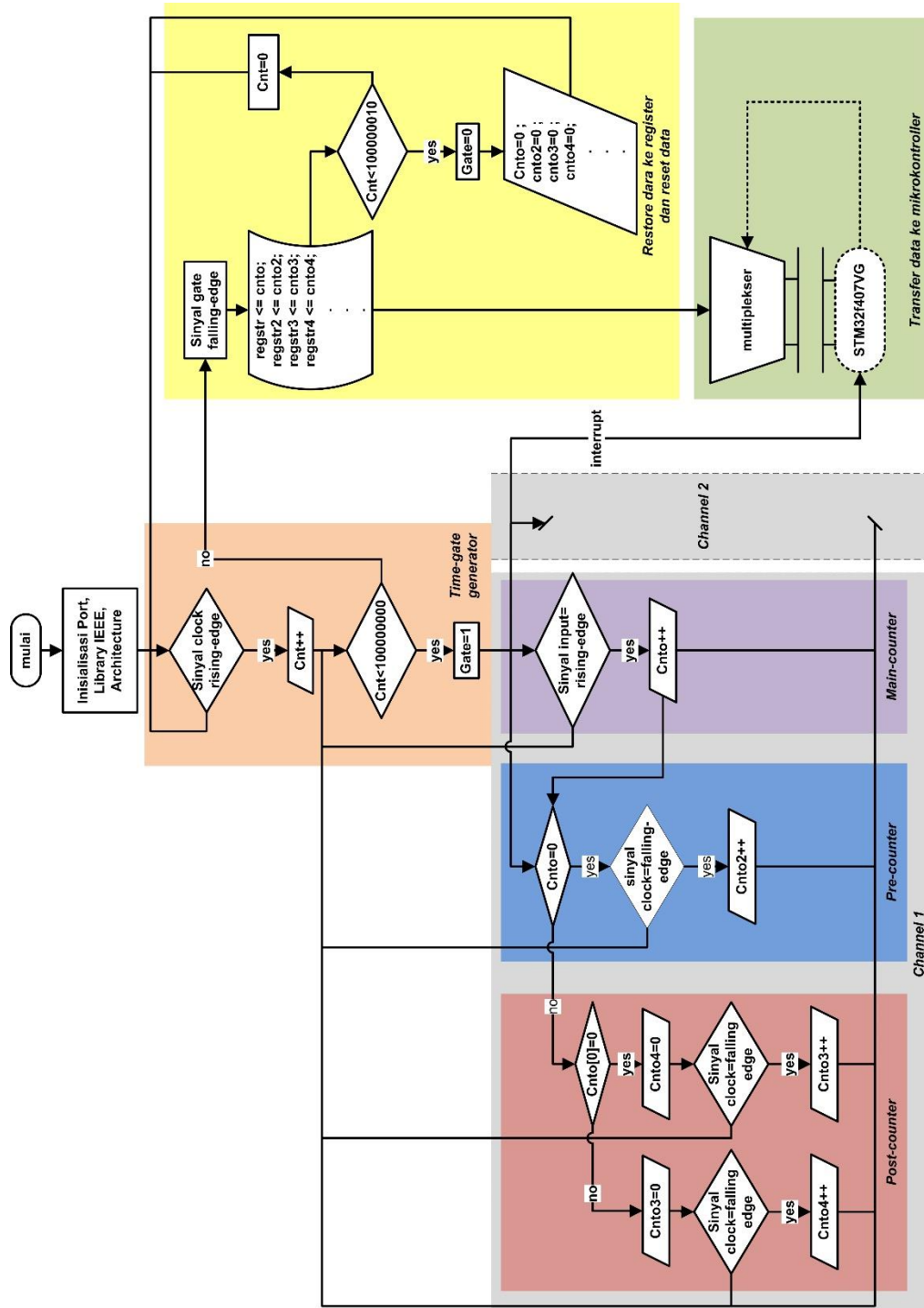
Setiap *counter* dikontrol oleh sinyal *time-gate* satu detik yang dibangkitkan oleh fungsi *time-gate generator* seperti yang dijelaskan pada paragraf sebelumnya. Fungsi *main-counter* bekerja seperti pada pencacah frekuensi metode konvensional, yaitu dengan cara mencacah transisi *rising-edge* dari sinyal input selama sinyal *time-gate* satu detik. *Main-counter* mampu mencacah hingga 26-bit atau setara dengan 67MHz.

Sedangkan fungsi *pre-counter* dan fungsi *post-counter* bekerja berdasarkan metode *reciprocal*, yaitu dengan mencacah *clock* eksternal OCXO 100MHz. Fungsi

pre-counter bekerja dengan mencacah *clock* eksternal OCXO selama durasi antara transisi *rising-edge* sinyal *time-gate* satu detik hingga transisi *rising-edge* pertama dari sinyal input. Cacahan dilakukan dengan cara menaikkan nilai variabel *c-pre-counter* setiap terjadinya transisi *falling-edge* sinyal *clock* eksternal. Sedangkan fungsi *post-counter* bekerja dengan cara mencacah *clock* eksternal OCXO selama durasi antara transisi *rising-edge* terakhir dari sinyal input (LSB-bit) hingga transisi *falling-edge* dari sinyal *time-gate* satu detik. Pada fungsi ini terdapat dua variabel cacahan (*cpostcountergenap* dan *cpostcounterganjil*) yang akan menyimpan data cacahan *clock* eksternal untuk satu gelombang penuh sinyal input. Variabel *cpostcountergenap* dan *cpostcounterganjil* masing-masing akan menyimpan nilai cacahan *clock* eksternal pada saat nilai cacahan sinyal input bernilai genap (LSB-bit=0) dan ganjil (LSB-bit=1). Variabel *cpostcountergenap* akan direset ketika terjadi perubahan nilai cacahan sinyal input dari genap ke ganjil dan sebaliknya variabel *cpostcounterganjil* akan direset ketika terjadi perubahan nilai cacahan sinyal input dari ganjil ke genap. Hal ini terus berlangsung secara sequensial hingga terdeteksi transisi *falling-edge* sinyal *time-gate* satu detik. Sehingga salah satu variabel ini (*cpostcountergenap* dan *cpostcounterganjil*) akan menyimpan nilai cacahan *clock* eksternal selama durasi antara transisi *rising-edge* terakhir dari sinyal input (LSB-bit) hingga transisi *falling-edge* dari sinyal *time-gate* satu detik. *Pre-counter* dan *post-counter* mampu mencacah hingga 10-bit atau 1023 cacahan. Nilai cacahan *precounter*, *postcounterganjil*, dan *postcountergenap* akan dijumlahkan sebelum dipindahkan ke register yang bersesuaian. Oleh karena itu, setiap *channel* akan memiliki data hasil cacahan yang tersimpan di register dengan kapasitas maksimum sebesar $26+12=38\text{bit}$.

Setiap *channel* terdapat dua variabel yang masing-masing akan menyimpan nilai cacahan *main-counter*, dan hasil penjumlahan *pre-counter*, *post-counter* genap, dan *postcounter* ganjil. Nilai cacahan dari masing-masing variabel ini akan dipindahkan ke *Register* ketika terjadi *falling-edge* sinyal *time-gate* satu detik sebelum dikirim ke mikrokontroller pada saat sinyal *time-gate* satu detik berikutnya. Setelah itu, variabel-variabel cacahan ini akan direset selama *time-delay* 100ns untuk digunakan sebagai variabel cacahan pada sinyal *time-gate* satu detik berikutnya. Selain itu, sinyal *time-gate* satu detik ini juga digunakan sebagai sinyal *interrupt* untuk mikrokontroller STM32F746NG yang menandakan bahwa CPLD siap mengirimkan data cacahan yang tersimpan pada *Register*. Pengiriman data dari CPLD ke mikrokontroller dijumpai oleh sebuah multiplexer 16-bit. Proses pengiriman dikontrol oleh paket data yang dikirim oleh mikrokontroller. Oleh karena itu, selama durasi *time-gate* satu detik ini terjadi proses cacahan dan pengiriman data cacahan satu detik sebelumnya. Proses pengiriman data akan dijelaskan lebih lanjut pada subbab 4.3.1.2.

Setelah membuat program yang sudah didesain tersebut, maka langkah selanjutnya adalah mensimulasikan program pencacah frekuensi tersebut untuk melihat kinerjanya. Hal ini dilakukan dengan memanfaatkan *software Isim Simulator* yang sudah terintegrasi pada *software Xilinx ISE Design Suite*.

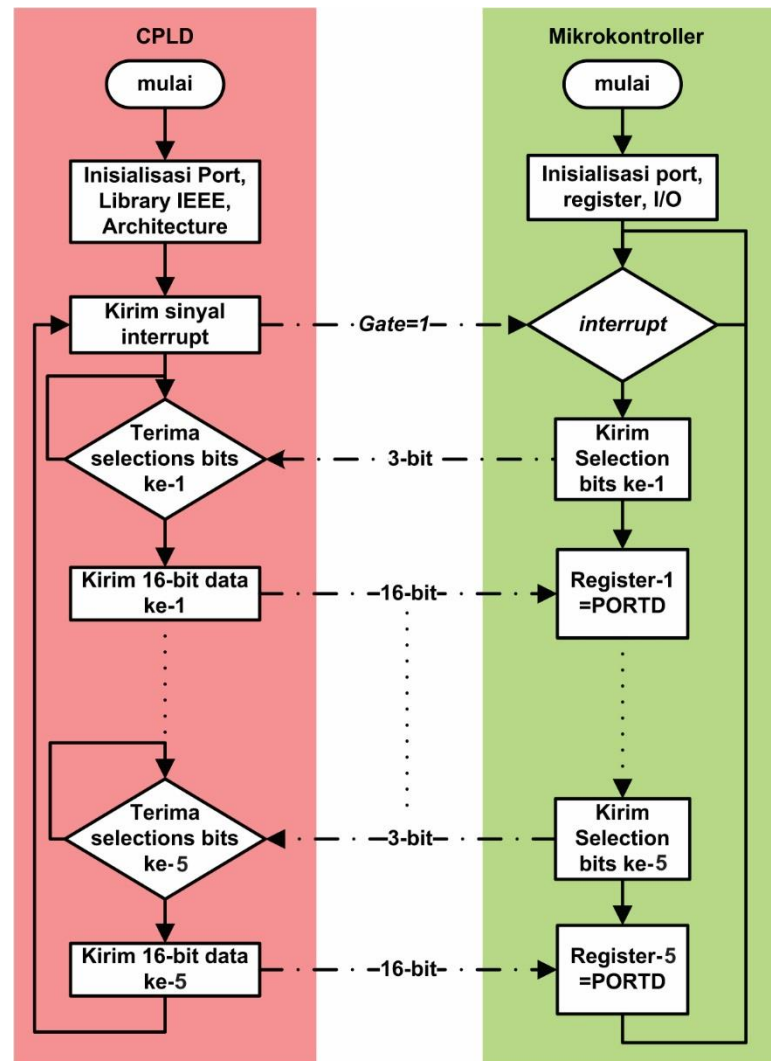


Gambar 4.4 Flowchart program pecah frekuensi

4.3.1.2 Interface CPLD dan STM32F746NG

Mikrokontroller STM32F746NG dalam penelitian ini merupakan sistem kontrol utama yang menangani beberapa fungsi. Salah satu fungsi utama dari mikrokontroller ini adalah menerima, menyimpan, dan mengolah data yang berasal dari CPLD. Oleh karena itu, dibutuhkan sistem komunikasi antara CPLD dan mikrokontroller. Sistem komunikasi yang dipilih adalah sistem komunikasi parallel. Sistem komunikasi ini memanfaatkan sebuah multiplekser yang dibangun di dalam CPLD. Multiplekser ini dirancang agar mampu mengirim paket data sebesar 16-bit setiap pengiriman.

Seperti yang dijelaskan pada subbab 4.3.1.1 bahwa kapasitas data setiap *channel* adalah sebesar 38-bit yang meliputi 26-bit data untuk *main-counter* dan 12-bit data hasil penjumlahan untuk 10-bit *pre-counter* dan 2x10-bit data untuk *post-counter*. Sehingga jumlah kapasitas data untuk kedua channel adalah sebesar 76-bit. Data ini akan dipecah menjadi 5x16-bit agar dapat dikirim secara parallel melalui multiplekser. Oleh karena itu, ukuran multiplekser yang dibangun di dalam CPLD adalah sebesar 80x16 dengan 3-bit selek. Alur program interface CPLD dan mikrokontroller dapat ditunjukkan pada Gambar 4.5. Pada *flowchart* tersebut dijelaskan bahwa ketika CPLD mengirimkan sinyal *interrupt* berupa sinyal *time-gate* satu detik ke mikrokontroller, maka mikrokontroller akan mengirimkan *selection bits* ke CPLD. *Selection bits* ini berukuran 3-bit yang akan memberikan kode pada multiplekser di dalam CPLD agar mengirimkan data berukuran 16-bit sesuai dengan kode yang dikirimkan tersebut. Oleh karena itu, di dalam CPLD itu sendiri harus terdapat fungsi yang mengidentifikasi kode yang dikirimkan mikrokontroller, sehingga CPLD akan mengirimkan data 16-bit sesuai dengan kode



Gambar 4.5 Flowchart interface CPLD dan Mikrokontroller STM32F746NG yang diterimanya. Kemudian mikrokontroller akan menerima data 16-bit tersebut melalui PORTD dan akan menyimpannya pada *Register* yang bersesuaian. Dalam keseluruhan proses, Mikrokontroller akan mengirimkan 5 kombinasi kode berukuran 3-bits dan CPLD akan mengirimkan 16-bit data setiap menerima 3-bit kode yang bersesuaian. Sehingga terjadi 5 kali proses pengiriman yang masing-masing berukuran 16-bit.

Setelah mikrokontroller STM32F746NG menerima 80-bit data hasil pengukuran dari CPLD, maka mikrokontroller akan mengkombinasikan data-data yang terkoleksi pada *Register* yang berbeda untuk mendapatkan 38-bit data untuk

masing-masing *channel* pengukuran. 38-bit data ini meliputi 26-bit data untuk *main-counter* dan 12-bit data hasil penjumlahan untuk 10-bit *pre-counter* dan 2x10-bit data untuk *post-counter*. Kemudian mikrokontroller akan mengolah data-data ini untuk mendapatkan nilai frekuensi yang terukur pada masing-masing *channel* dengan menggunakan persamaan (4.1)

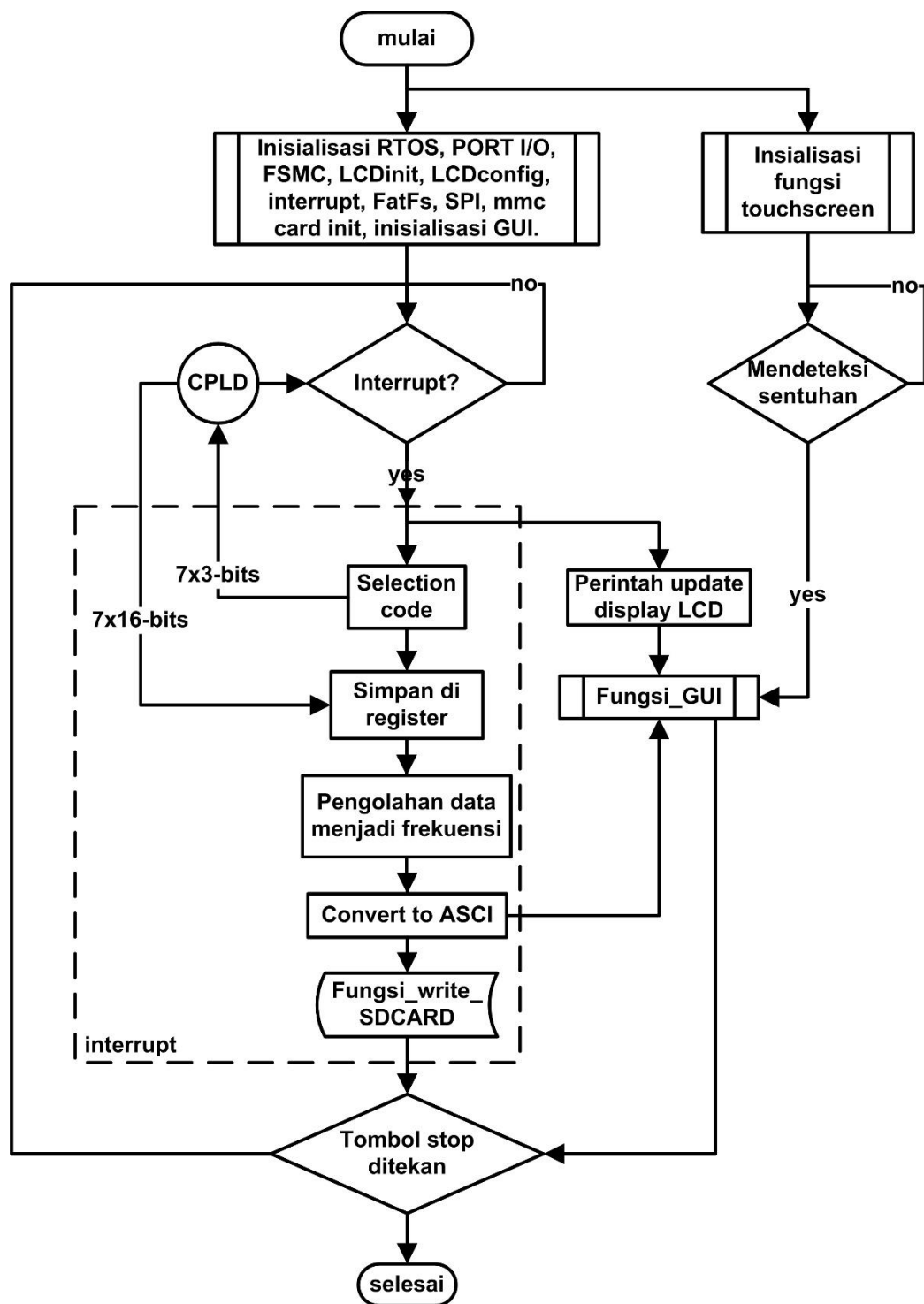
$$f = \frac{(r_{mc}-1)*100000000}{100000000-r_{sisa}} \quad (4.1)$$

Dimana f merupakan frekuensi (Hz), r_{mc} merupakan data 26-bit hasil cacahan pada *main-counter*, r_{sisa} merupakan data 12-bit data hasil penjumlahan untuk 10-bit *pre-counter* dan 2x10-bit data untuk *post-counter*.

4.3.1.3 Program Kontrol STM32F746NG

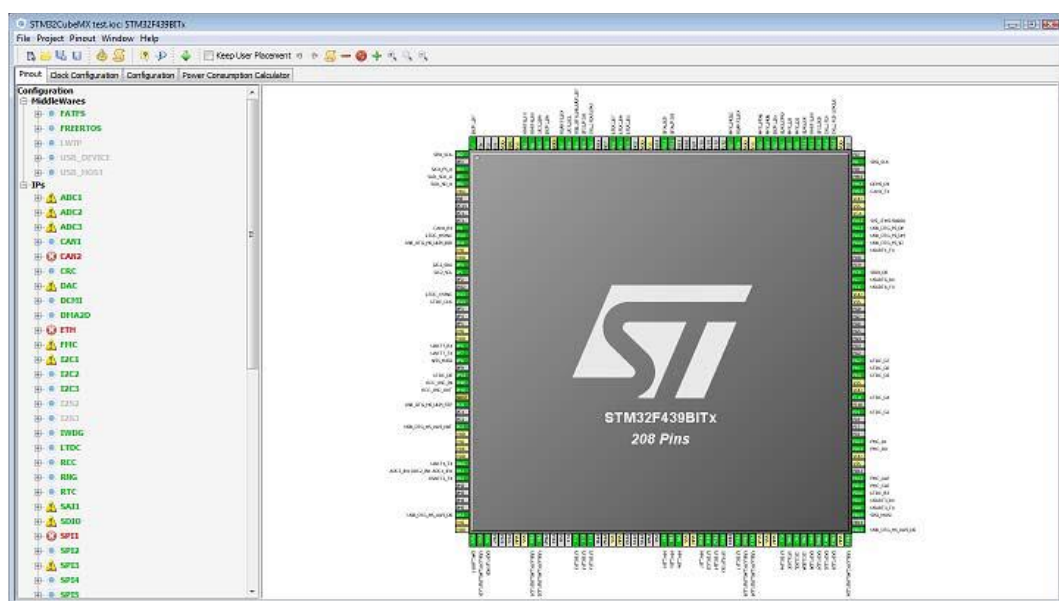
Secara garis besar, alur program yang ditanamkan di dalam mikrokontroller STM32F746NG dapat ditunjukkan seperti pada Gambar 4.6. Mikrokontroller ini merupakan komponen utama yang akan menangani beberapa fungsi seperti mengontrol LCD TFT Touchscreen dan microSD CARD Adapter, menerima dan mengolah data dari CPLD yang selanjutnya akan ditampilkan pada LCD TFT dan disimpan di microSD. Program dimulai dengan menginisialisasi seluruh komponen seperti library dan modul program yang dibutuhkan. Kemudian program akan mengecek apakah terdapat sinyal *interrupt* yang berasal dari CPLD *Frequency counter*. Mikrokontroller akan mengirimkan tujuh kombinasi data 3-bit apabila terdeteksi adanya sinyal *interrupt*. Kombinasi data 3-bit ini digunakan untuk memberikan perintah pada CPLD agar mengirimkan data 16-bit hasil pengukuran sesuai dengan kombinasi data 3-bit yang dikirimkan oleh mikrokontroller. Setelah itu, data-data yang dikirimkan oleh CPLD akan disimpan pada register yang

bersesuaian agar dapat diolah untuk mendapatkan nilai frekuensi berdasarkan persamaan (4.1). Nilai frekuensi ini dirubah ke dalam bentuk ASCII agar dapat disimpan ke dalam SD CARD melalui fungsi *write sd card*. Melalui fungsi *write sd card* ini, data frekuensi akan disimpan ke dalam sd card sesuai dengan waktu pencuplikannya. Hasil penyimpanannya akan berbentuk tabel yang terdiri dari waktu pencuplikan dan frekuensi hasil pengukuran untuk setiap *channel*. Selain itu, nilai frekuensi ini akan ditampilkan pada LCD TFT dalam bentuk grafik hubungan waktu pencuplikan dan frekuensi berdasarkan fungsi GUI (*Graphics User Interface*) yang dibangun di dalam mikrokontroller STM32F746NG. Fungsi GUI ini juga menerima output yang berasal dari sensor *touchscreen* sebagai media *user interface*. Namun, output dari sensor *touchscreen* ini terlebih dahulu diolah oleh fungsi *touchscreen* untuk mendapatkan nilai koordinat sensor *touchscreen* yang dikenai tekanan. Proses pencuplikan data pengukuran oleh mikrokontroller akan terus berlangsung hingga fungsi stop dijalankan dengan mengaktifkan fungsi stop yang berada di dalam fungsi GUI.



Gambar 4.6 Flowchart program pada Mikrokontroler STM32F746NG

Seluruh program yang dibangun di dalam mikrokontroller STM32F746NG ini memanfaatkan STM32Cube untuk memudahkan pengembangan sistem yang kompleks khususnya pada sistem *embedded*. Di dalam STM32Cube ini terdapat semua komponen dasar yang dibutuhkan untuk mengembangkan sebuah aplikasi yang berbasis mikrokontroller STM32. Salah satunya adalah STM32CubeMx (Gambar 4.7) yang digunakan untuk menkonfigurasi komponen, perhiperal, atau library yang dibutuhkan dalam membangun sebuah aplikasi dan kemudian membangkitkan sebuah kode program dasar sesuai dengan konfigurasi tersebut.

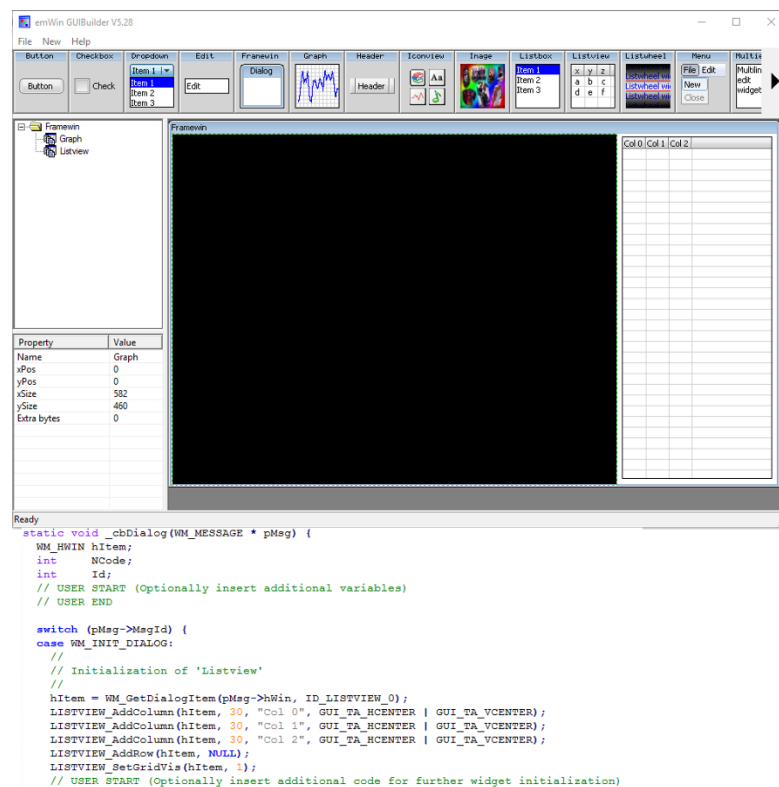


Gambar 4.7 STM32CubeMX

Salah satu komponen *middleware* yang terdapat di dalam STM32Cube adalah FreeRTOS yang merupakan salah satu RTOS(*Real Time Operating System*) yang bersifat *open source*. RTOS itu sendiri merupakan sebuah sistem operasi yang dioptimalkan untuk digunakan pada sistem *embedded* atau aplikasi *real-time*. Penggunaan RTOS pada sistem *embedded* akan meningkatkan respon program terhadap suatu *event*. Selain itu, setiap program yang dijalankan pada suatu aplikasi

dapat bekerja secara independen. Sehingga RTOS cocok digunakan untuk mengembangkan perangkat pencacah frekuensi portabel untuk QCM yang akan dilakukan pada penelitian ini. FreeRTOS merupakan salah satu RTOS yang didesain cukup ringan untuk dijalankan pada mikrokontroller, namun hal tersebut tidak membatasi penggunaannya pada mikrokontroller saja.

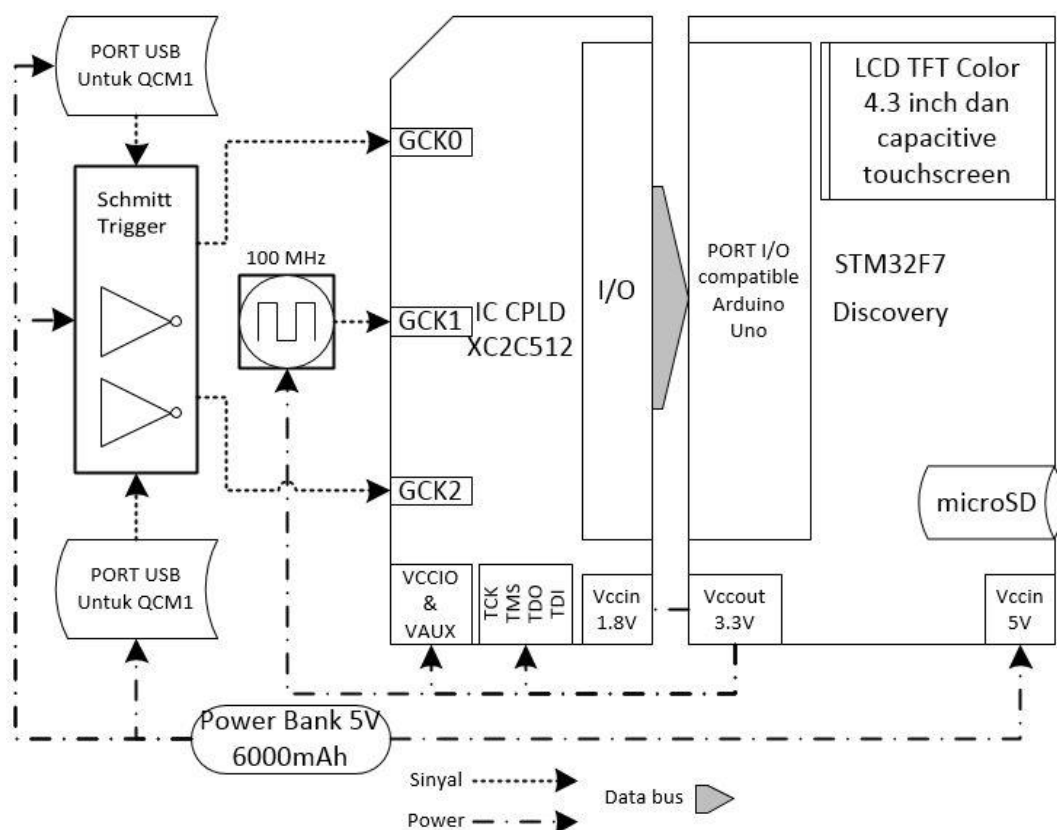
Graphics User Interface (GUI) yang akan ditampilkan pada LCD TFT dan media *user interface* dari sensor *Touchscreen* didesain menggunakan *Stemwin GUI Builder* seperti pada Gambar 4.8. Pada software ini terdapat modul GUI yang akan digunakan pada perangkat pencacah frekuensi portabel seperti modul *Graph* yang menampilkan grafik 2D.



Gambar 4.8 emWin GUIBuilder untuk mendesain program GUI

4.3.2 Desain Elektronik

Sistem pencacah frekuensi portabel untuk sensor QCM ini secara umum terdiri dari empat komponen utama, yaitu pencacah frekuensi berbasis CPLD XC2C512, modul STM32F7 Discovery berbasis mikrokontroler 32bit STM32F746NG dan sudah dilengkapi dengan LCD TFT *Touchscreen* dan microSD CARD adapter. Selain itu, terdapat sistem *power supply* yang terdiri dari *Powerbank* 6000 mAh output 5volt, dan regulator 1.8volt. Komponen-komponen ini terintegrasi untuk membentuk sebuah perangkat pencacah frekuensi portabel untuk sensor QCM dengan konsep seperti pada Gambar 4.9.

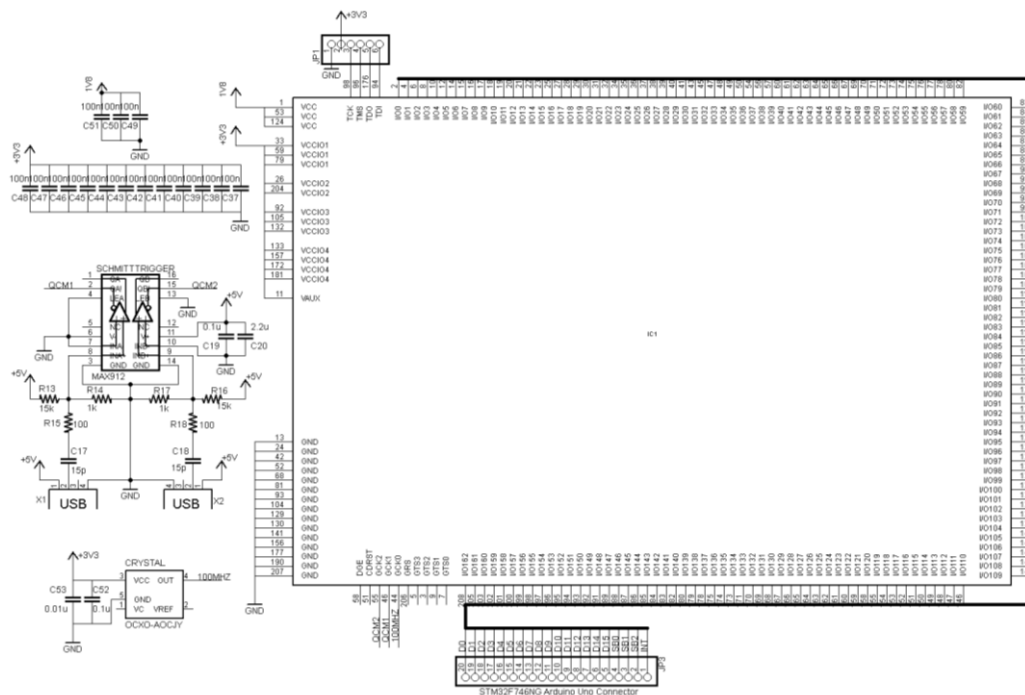


Gambar 4.9 Desain Elektronik Pencacah Frekuensi Portabel untuk Sensor QCM

4.3.2.1 Rangkaian CPLD Frequency Counter

Pencacah frekuensi pada penelitian ini dikembangkan menggunakan CPLD XC2C512. Board sistem CPLD XC2C512 itu sendiri didesain sesuai dengan desain program yang ditanamkan didalamnya seperti yang sudah dikemukakan pada subbab 4.3.1.1. Oleh karena itu, pada

Gambar 4.10 ditunjukkan skema rangkaian pencacah frekuensi yang dikembangkan pada penelitian ini. Rangkaian ini terdiri dari IC CPLD XC2C512, rangkaian *schmitt-trigger* berbasis MAX912, dan osilator OCXO 100MHz. Selain itu, terdapat dua port USB yang digunakan jalur yang menghubungkan antara output rangkaian osilator QCM dengan input rangkaian pencacah frekuensi berbasis CPLD. Selain itu, melalui port USB ini, sumber tegangan stabil digunakan untuk



mencatu rangkaian osilator QCM.

Gambar 4.10 Skema Rangkaian Pencacah Frekuensi Berbasis CPLD XC2C512

Rangkaian *schmitt-trigger* yang digunakan pada penelitian ini merupakan rangkaian yang dikembangkan oleh Wijayanto (Wijayanto, 2014). Penggunaan

schmitt-trigger dimaksudkan untuk memfilter sinyal sensor QCM sebelum dicacah oleh CPLD. Hal ini dilakukan dengan cara memfilter amplitudo dari sinyal sensor QCM. Memfilter amplitudo itu sendiri dilakukan dengan cara memberikan batas *hysteresis* pada amplitudo input yang masuk. Pemberian batas *hysteresis* ini dapat dilakukan dengan menggunakan rangkaian *schmitt trigger* yang memiliki dua batas *threshold*. Pada skema rangkaian tersebut, *hysteresis* dari *schmitt trigger* adalah sebesar 0.5 Volt dengan batas *threshold* bawah sebesar 0.5volt dan batas *threshold* atas sebesar 1volt. Sinyal input yang tidak dapat melewati batas *hysteresis* ini akan dianggap noise dan tidak akan diloloskan. Oleh karena itu, penggunaan batas *threshold* sebesar 0.5 volt sudah cukup untuk memfilter sinyal dari sensor QCM yang diketahui memiliki rentang tegangan 2volt pada saat pembebanan dan 6volt pada saat tanpa pembebanan. Rangkaian *schmitt trigger* ini tidak hanya memfilter *noise*, tetapi juga mengubah input sinyal analog menjadi sinyal digital dengan level TTL yang berbalik fase 180⁰ (Wijayanto, 2014).

Berdasarkan program yang akan ditanamkan ke dalam CPLD, maka desain rangkaian CPLD terdiri dari 2 input untuk sensor QCM, jalur komunikasi dengan Mikrokontroller yang terdiri 16 output data, 3 input *selections-bits*, dan 1 output interrupt, serta 1 input untuk osilator eksternal OCXO 100MHz. IC CPLD XC2C512 membutuhkan catu daya 1.8Volt yang terhubung ke pin Vcc. Selain itu, terdapat empat *power bank* yang mencatu tiap pin I/O pada CPLD yang terhubung ke pin V_{CCIO1}, V_{CCIO2}, V_{CCIO3}, dan V_{CCIO4}. Setiap *power bank* ini terhubung secara *stande alone* dengan rentang tegangan yang dapat dipilih berkisar antara 1.4 Volt hingga 3.6 Volt (XILINX, 2007). Oleh karena itu, pada penelitian ini dipilih tegangan sebesar 3.3volt sebagai catu daya keempat powerbank tersebut yang

disesuaikan dengan rentang tegangan pada port I/O mikrokontroler STM32F407VG. Selain itu, setiap pin yang terhubung dengan sumber tegangan pada CPLD diparalel sedekat mungkin dengan kapasitor 100nF untuk mengurangi noise.

Pada rangkaian pencacah frekuensi berbasis CPLD ini juga terdapat osilator OCXO 100MHz sebagai *clock* eksternal bagi CPLD. Osilator OCXO yang digunakan memiliki karakteristik frekuensi output 100MHz CMOS dengan stabilitas frekuensi $\pm 5\text{ppb}$ pada rentang suhu 0°C hingga 50°C dan membutuhkan daya 3.3Volt (Abracon, 2008). Output dari osilator ini terhubung dengan CPLD melalui pin Global *clock* 1 (GCK0) yang diatur sebagai input. Penggunaan pin ini sebagai input bagi osilator OCXO dikarenakan pin ini memiliki nilai *propagation delay* yang paling rendah dibandingkan dengan port yang lain (Wijayanto, 2014). Dengan alasan yang sama, maka pin GCK1 dan GCK2 masing-masing digunakan sebagai masukan sensor QCM1 dan QCM2. Pada rangkain tersebut juga disiapkan port untuk JTAG sebagai jalur yang digunakan untuk mengupload program ke CPLD.

4.4 Pengujian Alat

Pengujian dilakukan untuk menguji performa dari masing-masing komponen penyusun perangkat pencacah frekuensi portabel untuk sensor QCM. Tidak hanya itu, pengujian juga dilakukan untuk melihat performa alat ketika digunakan untuk mengukur frekuensi sensor QCM secara portabel. Pangujian dilakukan baik melalui simulasi program maupun rangkaian elektronik. Pengujian melalui simulasi program dilakukan untuk mendapatkan desain program terbaik sebelum diimplementasikan ke dalam rangkaian elektronik. Sehingga memudahkan

dalam mendesain skema rangkaian yang disesuaikan dengan desain program tersebut.

4.4.1 Pengujian CPLD *Frequency Counter* dan *Interface* dengan STM32F746NG

Pengujian terhadap CPLD *Frequency Counter* dilakukan baik melalui simulasi program maupun pengujian terhadap rangkaian CPLD *Frequency Counter* itu sendiri. Tahap awal dalam pengujian ini melalui simulasi program menggunakan *software Isim Simulator* yang sudah terintegrasi pada *software Xilinx ISE Design Suite*. Pengujian ini khususnya dilakukan untuk melihat alur kerja program dan karakteristik program yang dibangun (Wijayanto, 2014). Setiap bagian program yang terdiri dari *time-gate generator*, *main-counter*, *pre-counter*, *post-counter*, *Register*, dan multiplexer akan diuji. Simulasi ini akan menunjukkan karakteristik pembentukan *time-gate* satu detik oleh *time-gate generator*. Selain itu, simulasi ini juga akan menunjukkan proses pencacahan oleh *main-counter*, *pre-counter*, dan *post-counter* selama *time-gate* satu detik tersebut. Tidak hanya itu, simulasi ini juga akan memberikan gambaran bagaimana proses pemindahan data cacahan ke *Register* selama *time-delay* 100ns, serta menunjukkan bagaimana proses kerja multiplexer dalam mengirimkan data berdasarkan *selection-bits* yang diterimanya selama *time-gate* satu detik. Setelah simulasi menunjukkan bahwa setiap bagian program memperlihatkan kinerja sesuai yang diinginkan, maka selanjutnya adalah mensimulasikan program secara keseluruhan dengan memberikan input dengan karakteristik yang divariasikan seperti frekuensi dan kemudian melihat output dari program tersebut. Salah satu manfaat dari pengujian ini adalah melihat akurasi dan

resolusi pencacah frekuensi secara program. Langkah selanjutnya adalah mengaplikasikan program tersebut ke dalam IC CPLD.

Setelah melakukan pengujian terhadap program pencacah frekuensi, maka langkah selanjutnya adalah mengaplikasikan program tersebut ke dalam IC CPLD. Kemudian melakukan beberapa pengujian untuk melihat karakteristik dan kinerja dari IC CPLD yang sudah diprogram ini. Akan tetapi, pengujian terhadap IC CPLD ini harus melibatkan STM32F746NG yang berfungsi mengolah hasil cacahan oleh IC CPLD dan menampilkannya di layar monitor. Oleh karena itu, perlu dibangun sistem komunikasi antara CPLD dan STM32F746NG seperti yang diutarakan pada subbab 4.3.1.2. Dalam tahap pengujian ini, hasil cacahan yang sudah diolah oleh STM32F746NG tidak secara langsung ditampilkan di LCD TFT, namun ditampilkan di layar monitor PC. Oleh karena itu, perlu disiapkan sistem komunikasi antara STM32F746NG ini dengan PC. Sistem komunikasi yang dipilih adalah sistem komunikasi USB HID yang dibangun didalam STM32F746NG. Melalui sistem komunikasi ini, data cacahan yang sudah diolah oleh STM32F746NG akan dikirim ke PC dan ditampilkan pada monitor.

Setelah itu, langkah selanjutnya adalah menguji pencacah frekuensi berbasis CPLD ini untuk mengetahui karakteristik, seperti akurasi dan resolusinya. Pengujian dilakukan dengan memberikan input yang memiliki frekuensi yang sudah terkalibrasi dan melihat hasil pengukuran yang ditampilkan di monitor. Monitor akan menampilkan frekuensi terukur hasil pengolahan oleh mikrokontroller untuk melihat akurasi, presisi, dan resolusi pencacah frekuensi berbasis CPLD ini.

Pencacah frekuensi berbasis CPLD ini dikhususkan untuk mengukur frekuensi sensor QCM, sehingga perlu dilakukan pengujian pengukuran dengan menggunakan sensor QCM. Pengujian dilakukan dalam dua kondisi, yaitu kondisi tanpa pembebanan dan kondisi dengan pembebanan untuk melihat respon dari pencacah frekuensi berbasis CPLD tersebut. Sensor QCM yang digunakan adalah kristal kuarsa dengan frekuensi 10MHz yang menggunakan elektroda perak.

4.4.2 Pengujian *Graphics User Interface* (GUI)

Setelah dilakukan pengujian terhadap pencacah frekuensi berbasis CPLD yang sudah dikemukakan pada subbab 4.4.1, langkah selanjutnya adalah menguji *Graphics User Interface* (GUI) pada LCD TFT *Touchscreen* yang sudah terhubung dengan STM32F746NG. Salah satu tujuan pengujian ini adalah mensinkronkan hasil pengukuran pencacah frekuensi dengan nilai yang ditampilkan pada LCD TFT, baik dari segi grafik maupun nilai frekuensi terukur yang ditampilkan secara *real-time*. Pengujian dilakukan dengan menampilkan hasil pengukuran di layar monitor maupun di LCD TFT secara simultan. Kemudian diamati bentuk grafik maupun nilai yang ditampilkan pada kedua penampil tersebut.

4.4.3 Pengujian Interface micro SD Adapter dan STM32F746NG

Micro SD merupakan perangkat yang digunakan untuk menyimpan hasil pengukuran agar dapat dianalisis lebih lanjut. Micro SD ini terhubung ke mikrokontroler STM32F746NG melalui komunikasi SDMMC. Dalam sistem komunikasi ini mikrokontroler berperan sebagai *master* dan micro SD berperan sebagai *slave*. Dalam sistem komunikasi tersebut terdapat fungsi *write* dan *read* yang masing-masing berfungsi menyimpan hasil pengukuran dan membaca data

yang tersimpan pada micro SD. Oleh karena itu, perlu dilakukan pengujian terhadap sistem komunikasi antara mikrokontroller dan micro SD ini. Pengujian dilakukan dengan menyimpan data hasil pengukuran dan membaca kembali hasil pengukuran tersebut di komputer.

4.4.4 Pengujian Sistem Secara Keseluruhan Menggunakan sensor QCM

Pengujian ini dilakukan untuk melihat kinerja dari pencacah frekuensi portabel ketika digunakan untuk mengukur frekuensi sensor QCM secara parallel. Hal yang perlu dilakukan sebelum pengujian adalah menghubungkan pencacah frekuensi portabel ini dengan rangkaian osilator. Sensor QCM ditempatkan sebagai resonator pada rangkaian osilator ini, sehingga output dari rangkaian osilator ini merupakan frekuensi resonansi dari sensor QCM itu sendiri. Sensor QCM yang digunakan berupa kristal kuarsa mode AT-cut yang memiliki frekuensi resonansi 10MHz dengan elektroda perak. Kristal kuarsa ini ditempatkan dalam sebuah konstruksi sel selama pengujian berlangsung.

Pengujian dilakukan dengan memberikan beberapa perlakuan terhadap kristal kuarsa yang digunakan. Kemudian pencacah frekuensi portabel yang akan dikembangkan pada penelitian ini akan mengukur frekuensi kristal kuarsa untuk masing-masing perlakuan tersebut. Hasil pengukurannya akan ditampilkan dalam bentuk grafik dan nilai secara *real-time* pada LCD TFT. Hasil pengukuran tersebut juga disimpan pada microSD. Selain itu, pengujian dilakukan untuk mengetahui kemampuan LCD TFT untuk menunjukkan perbedaan hasil pengukuran dalam bentuk grafik untuk perlakuan yang berbeda-beda.